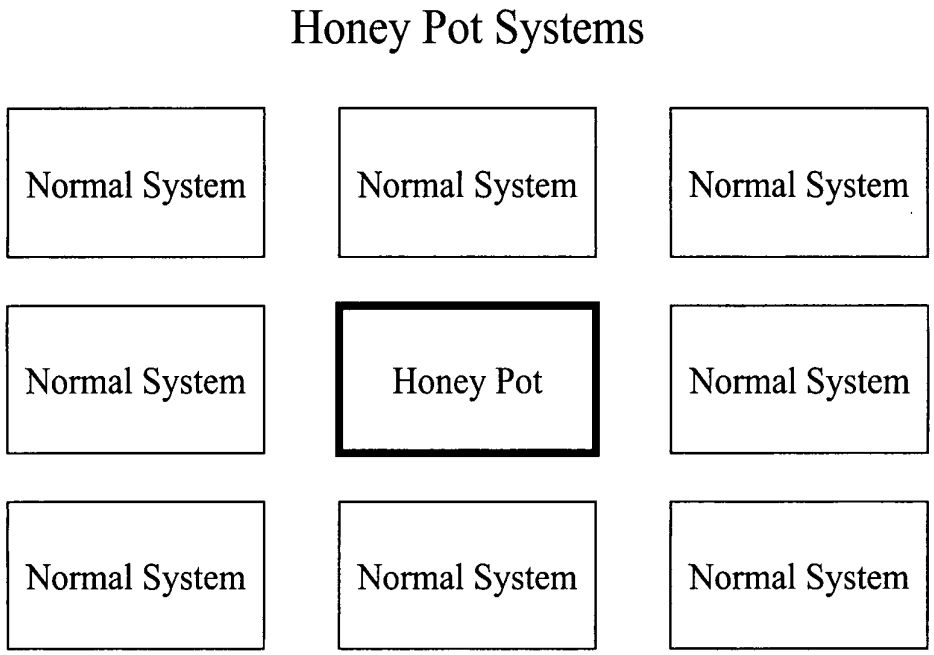


Fig. 1A
(Prior Art)

Proxy servers in firewalls and standard anonymizer services

- A surrogate for the real thing
- Proxy services
  - Take requests from clients
  - Translate for servers
  - Take responses from servers
  - Translate for clients

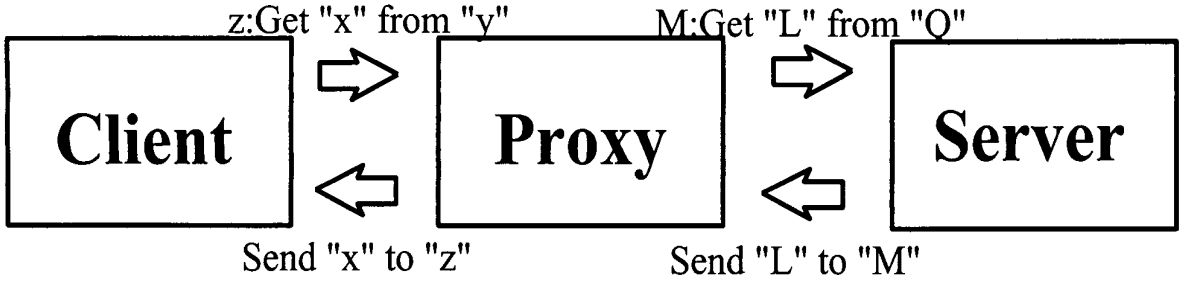


Fig. 1B
(Prior Art)

Front-end back-end firewall systems

- Front end in the firewall
- Back end in the internal network
- Communications limited (router)
- Addresses are not translated
  - intermediate machines use other machines

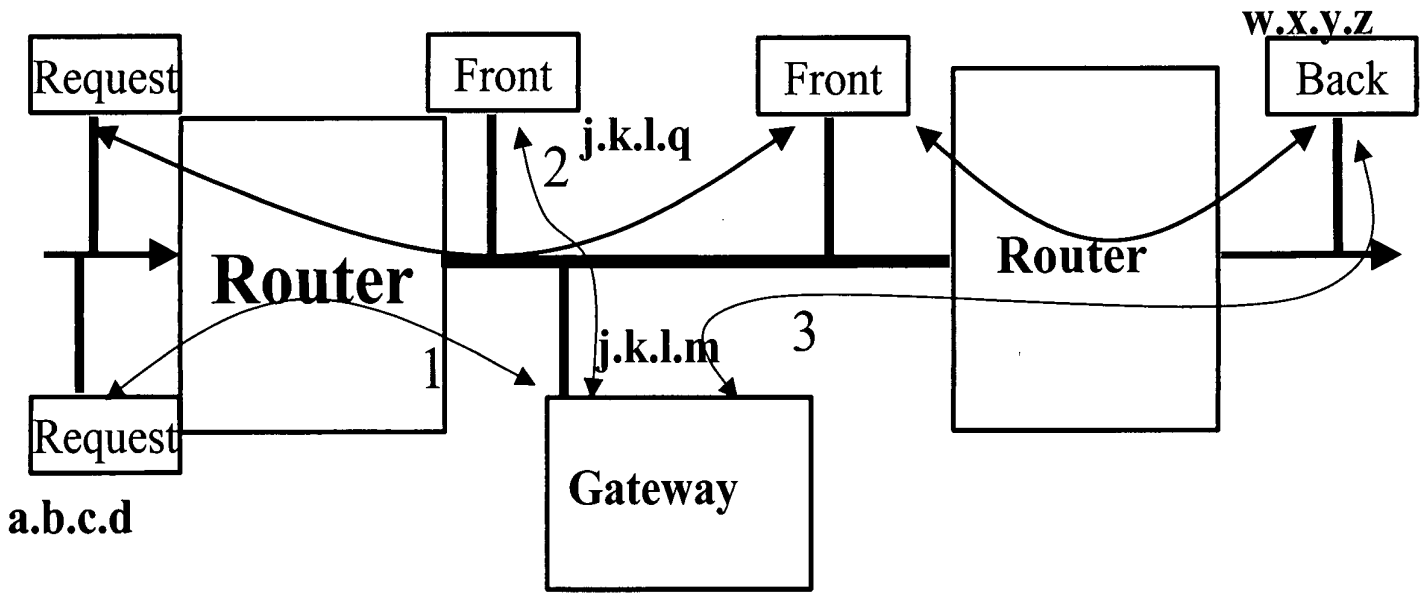


Fig. 1C
(Prior Art)

Original Deception ToolKit

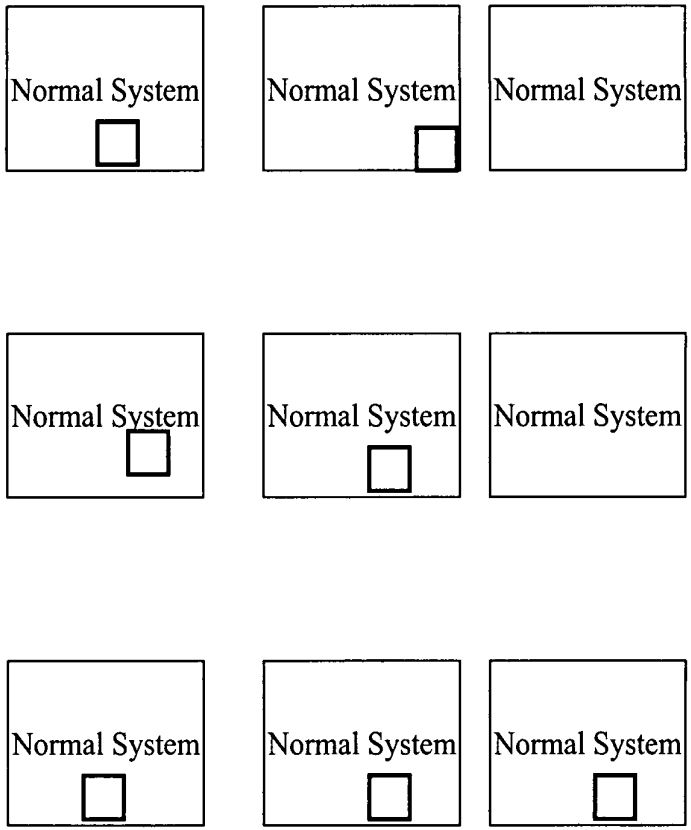


Fig. 2

Multiple Deceptions in One Box

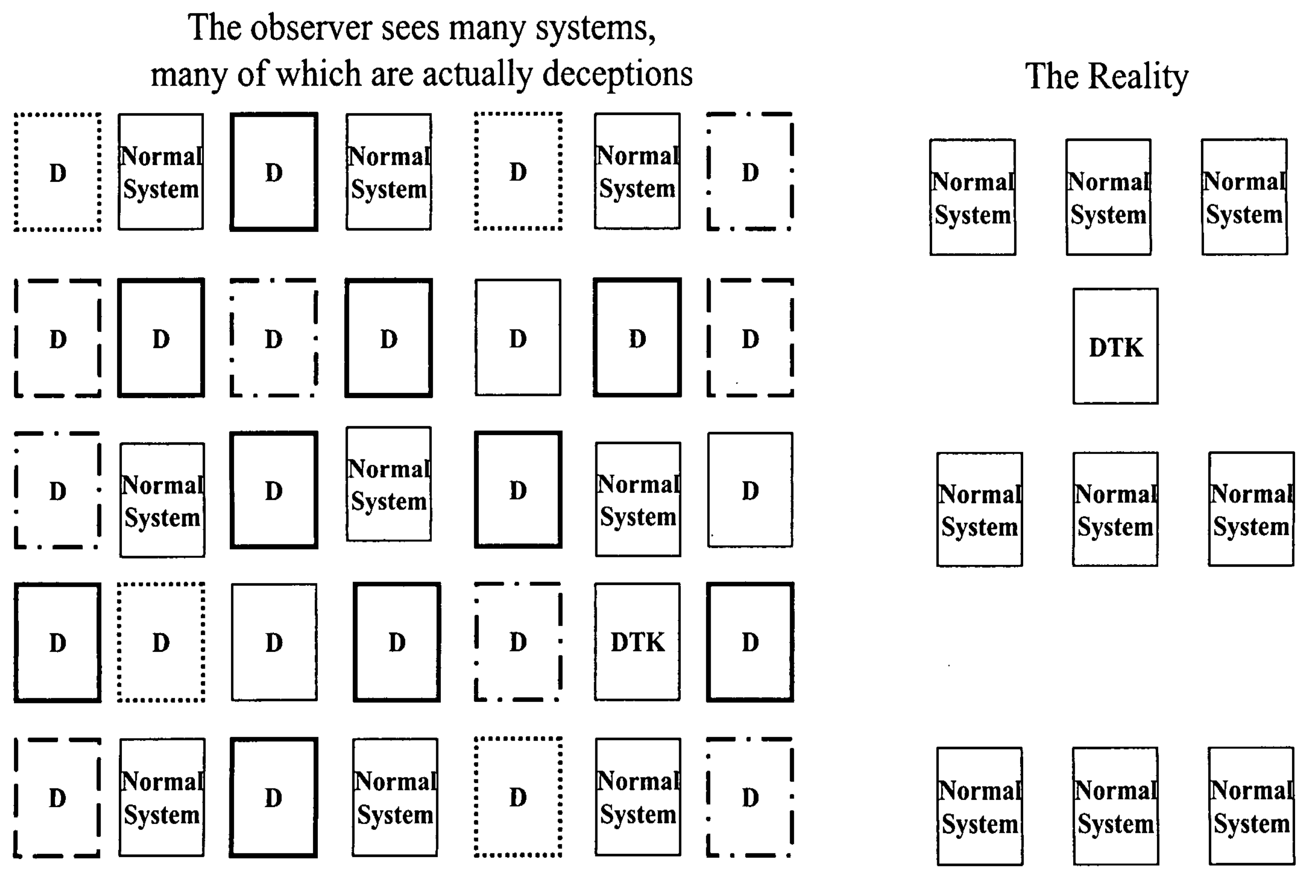
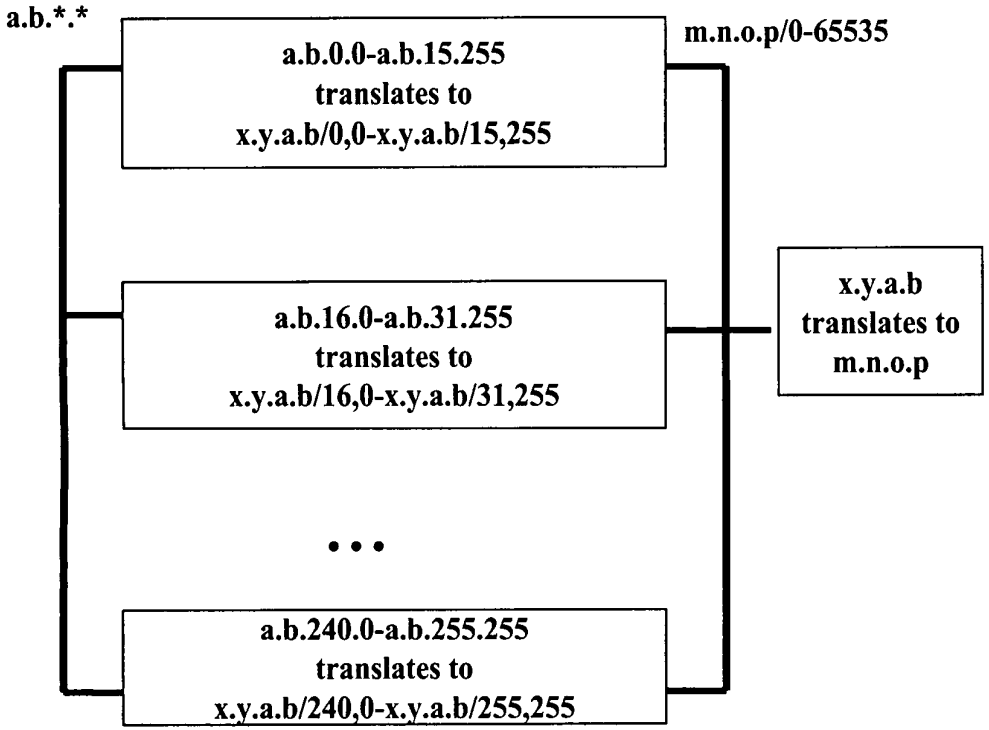


Fig. 3

Two Example Translation Designs

Using current technology, a two step system can translate for 64,000 addresses using 17 computers



Using a single system, the same translation can be done in software or with custom hardware if very high speed is desired

Input is examined using 'promiscuous mode' and a translation table is used to associate each internal address to an external address/port number

Fig. 4A

Multiple Address Translations

Addresses are translated multiple times to allow deception networks to be separated from normal networks, to allow 'real' machines to replace low fidelity deceptions, and to allow increased indirection & obscurity

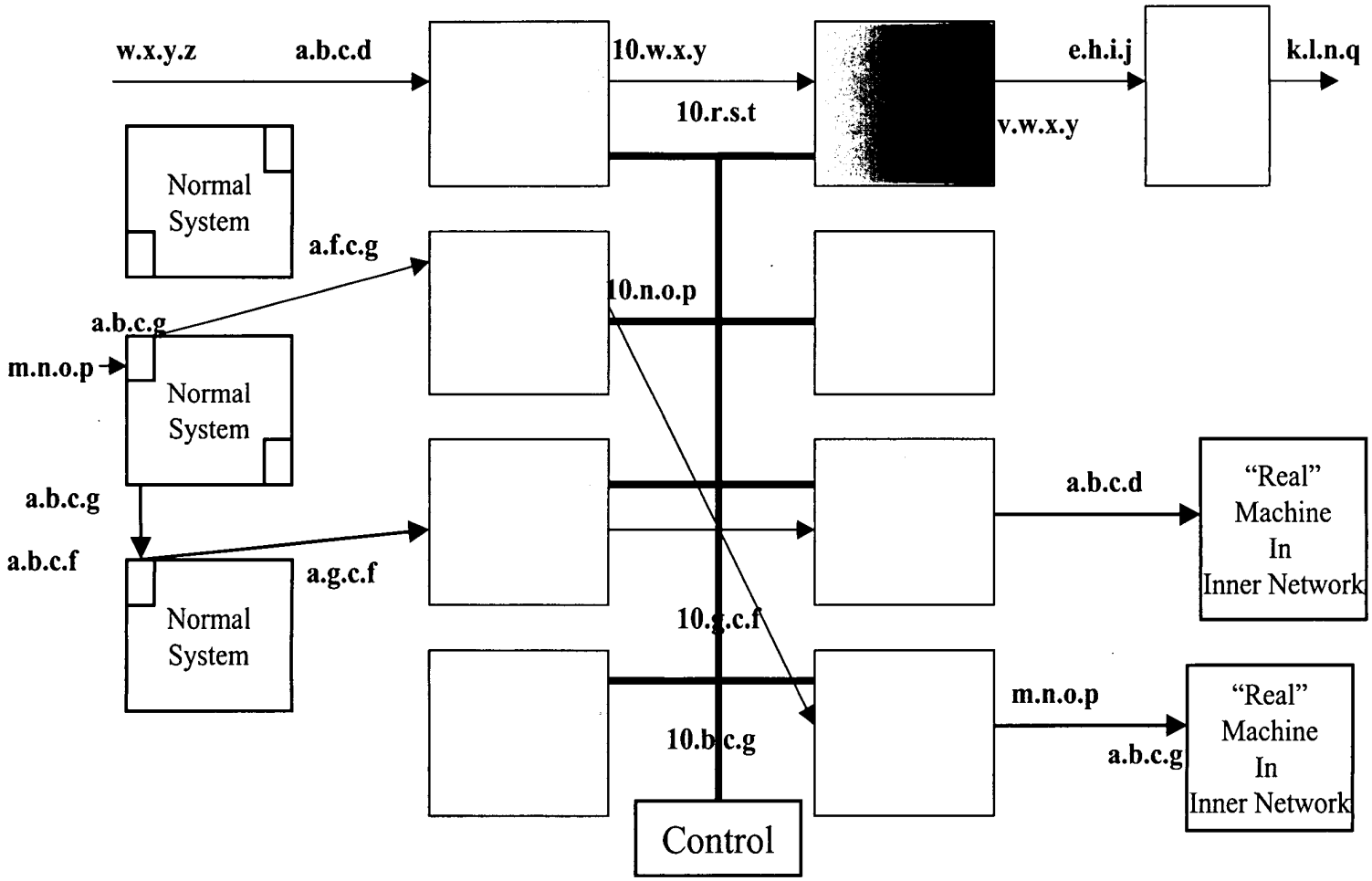


Fig. 4B

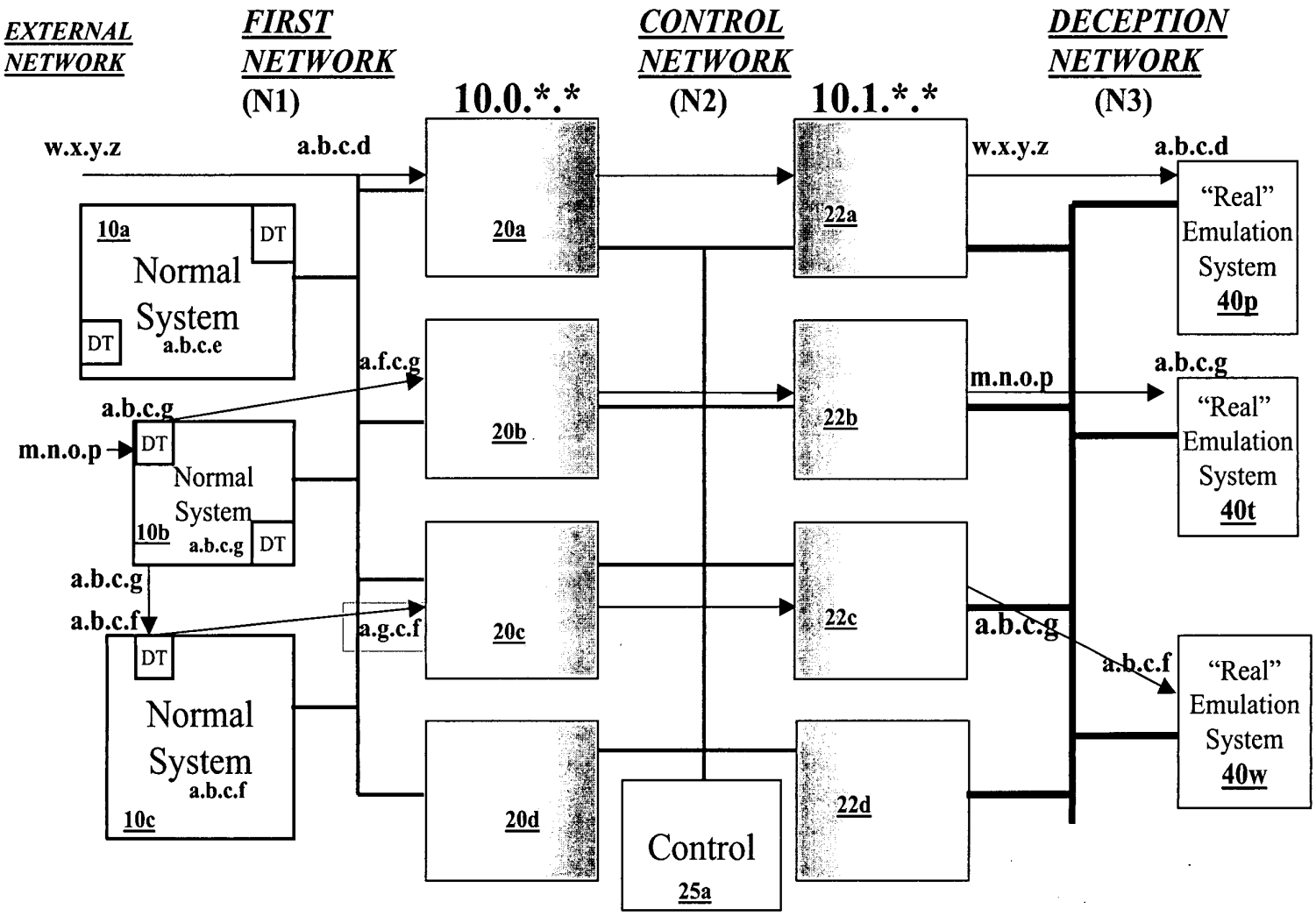


Fig. 4C

What the Student Sees

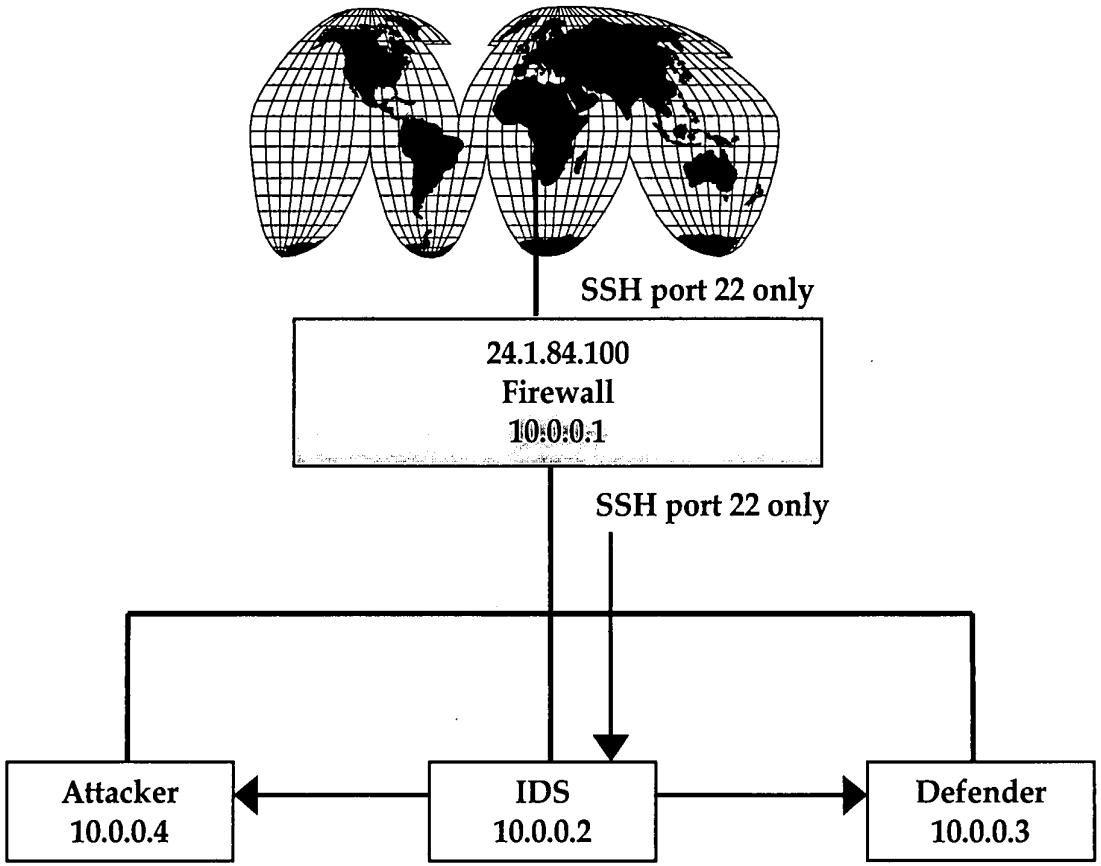


Fig. 5A

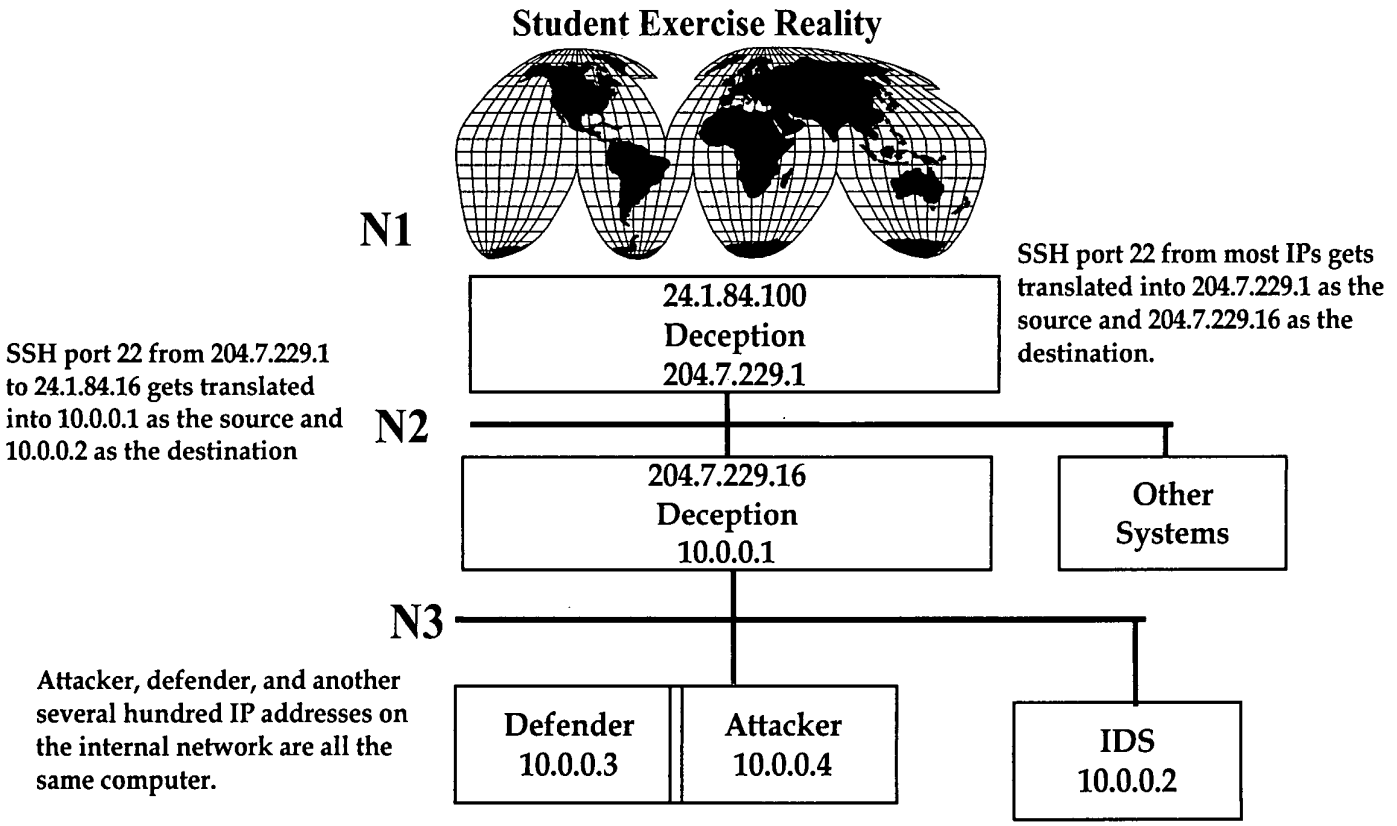
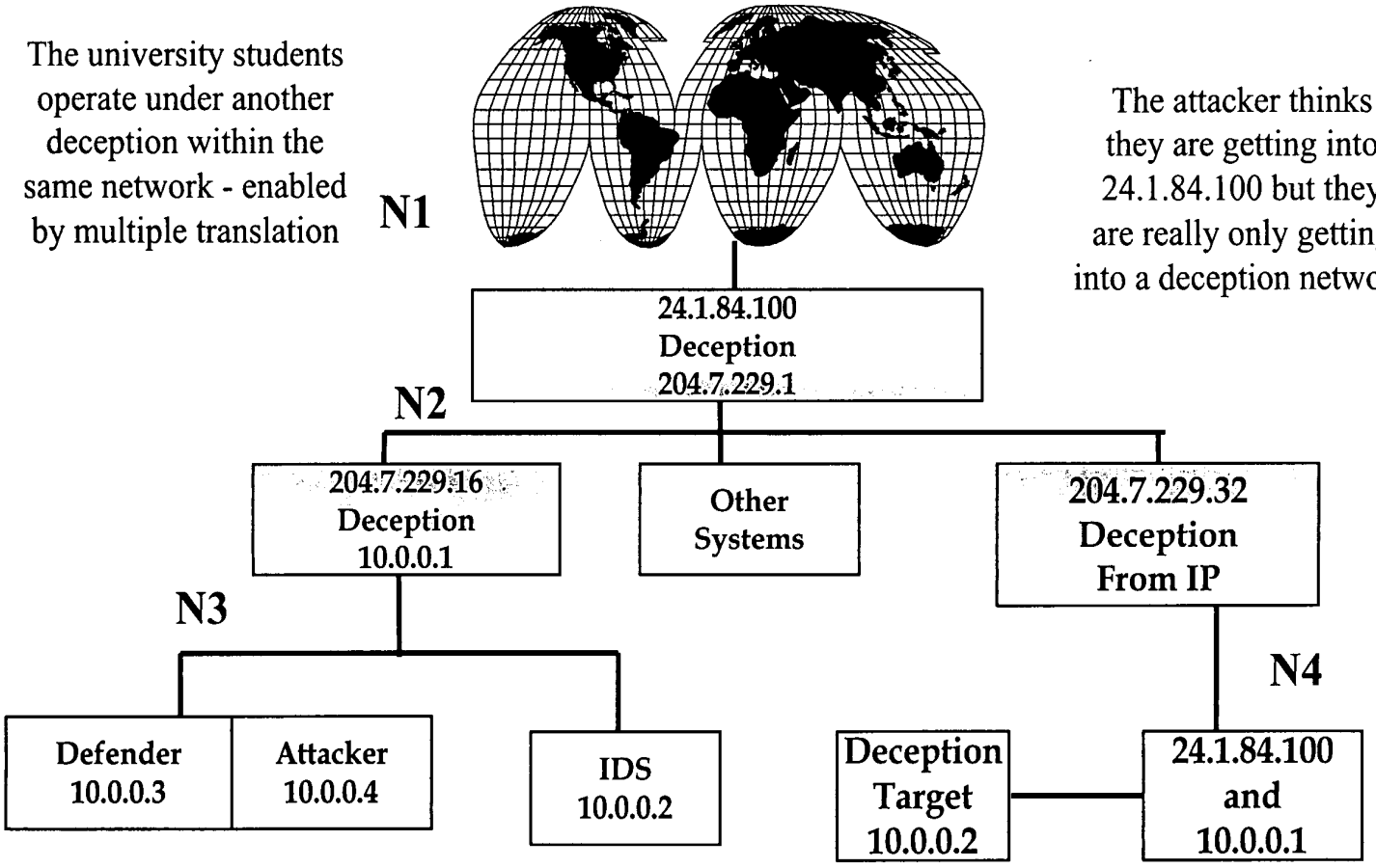


Fig. 5B

Another Example

The university students operate under another deception within the same network - enabled by multiple translation



The attacker thinks they are getting into 24.1.84.100 but they are really only getting into a deception network

Fig. 5C

Redirection and Obscure Request

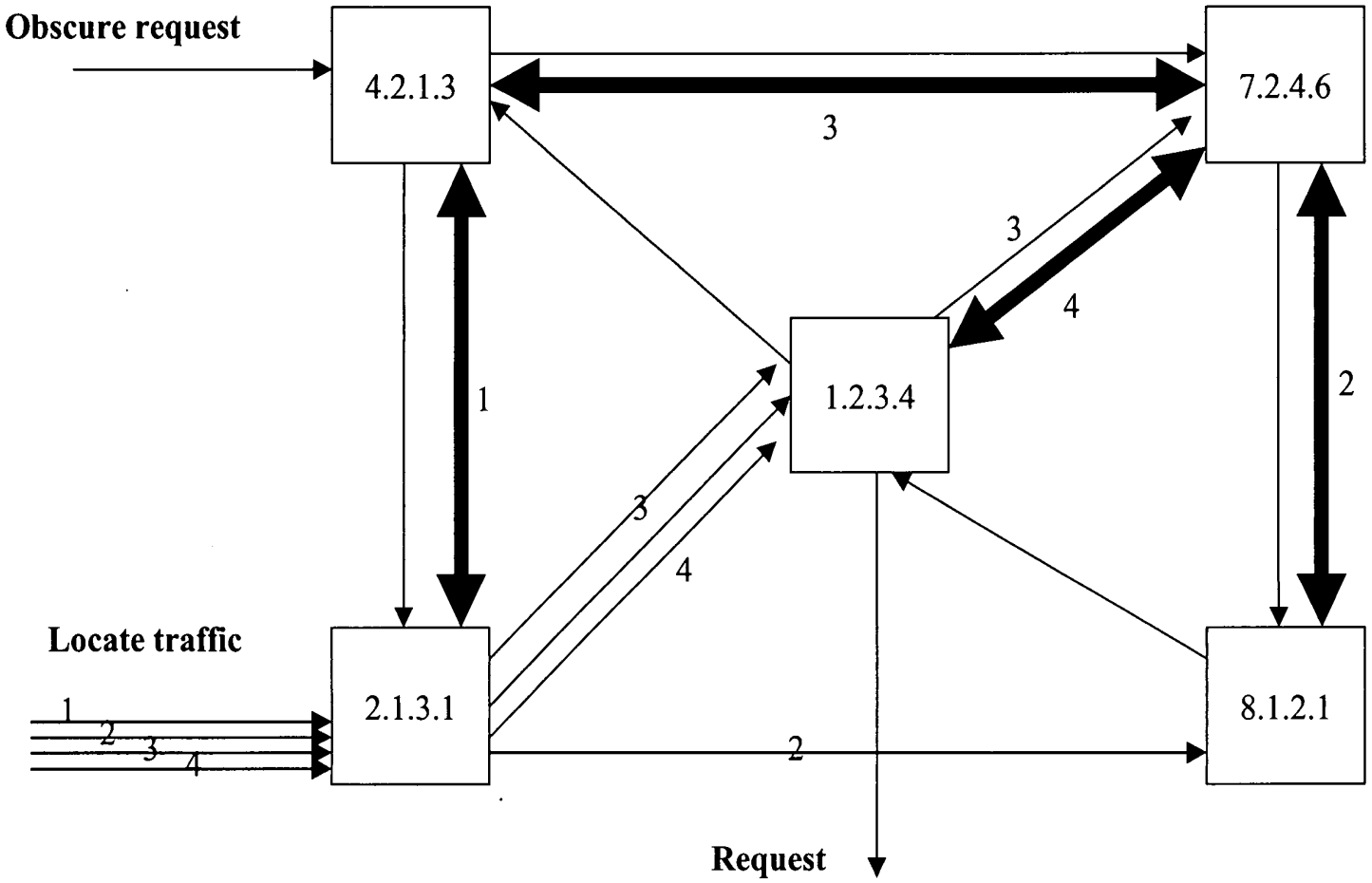


Fig. 6

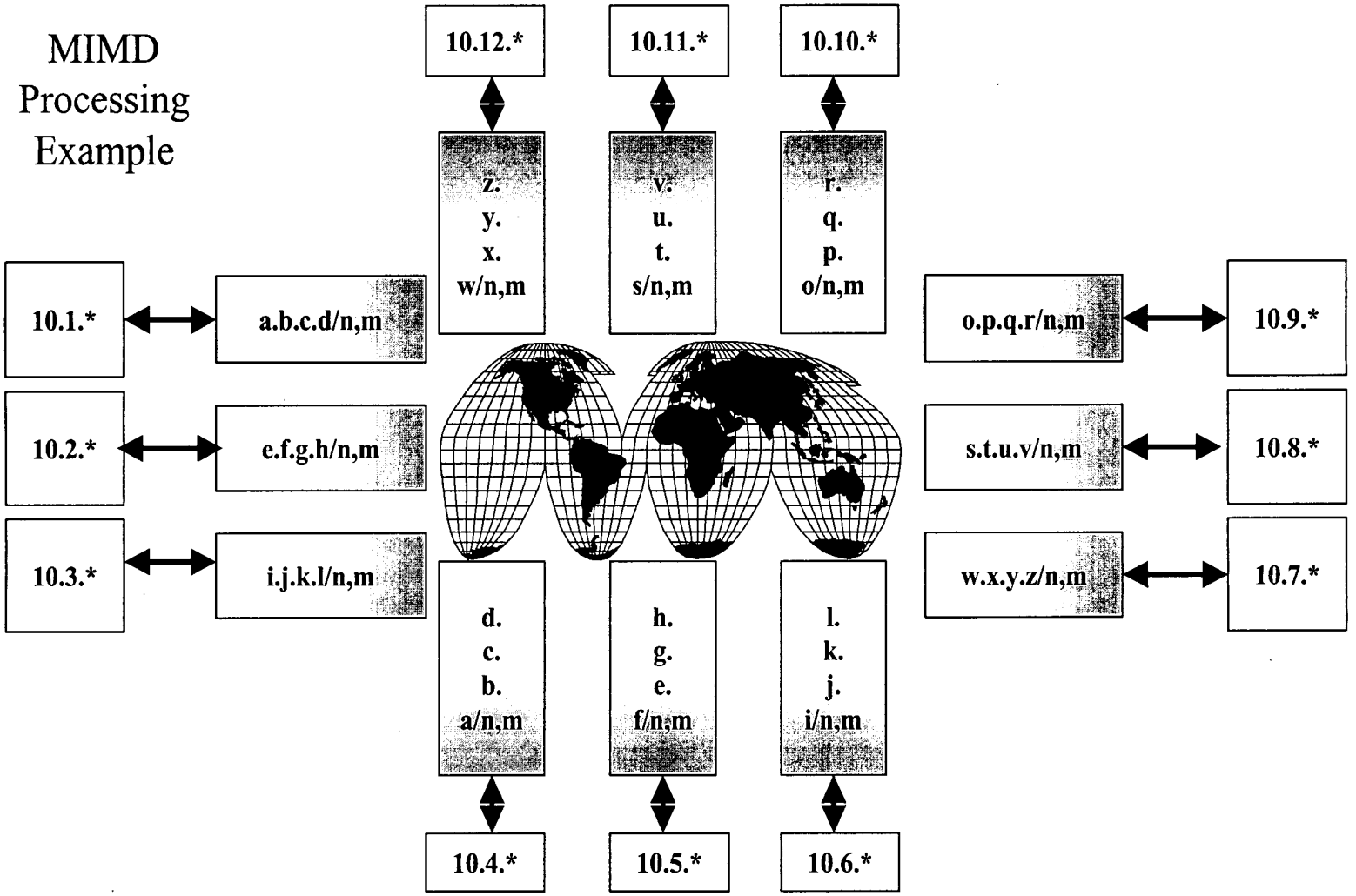

MIMD
Processing
Example
10.12.*
10.11.*
10.10.*
z.
y.
x.
w/n,m
v.
u.
t.
s/n,m
r.
q.
p.
o/n,m
10.1.*
a.b.c.d/n,m
10.2.*
e.f.g.h/n,m
10.3.*
i.j.k.l/n,m
o.p.q.r/n,m
10.9.*
s.t.u.v/n,m
10.8.*
w.x.y.z/n,m
10.7.*
d.
c.
b.
a/n,m
h.
g.
e.
f/n,m
l.
k.
j.
i/n,m
10.4.*
10.5.*
10.6.*

Fig. 7

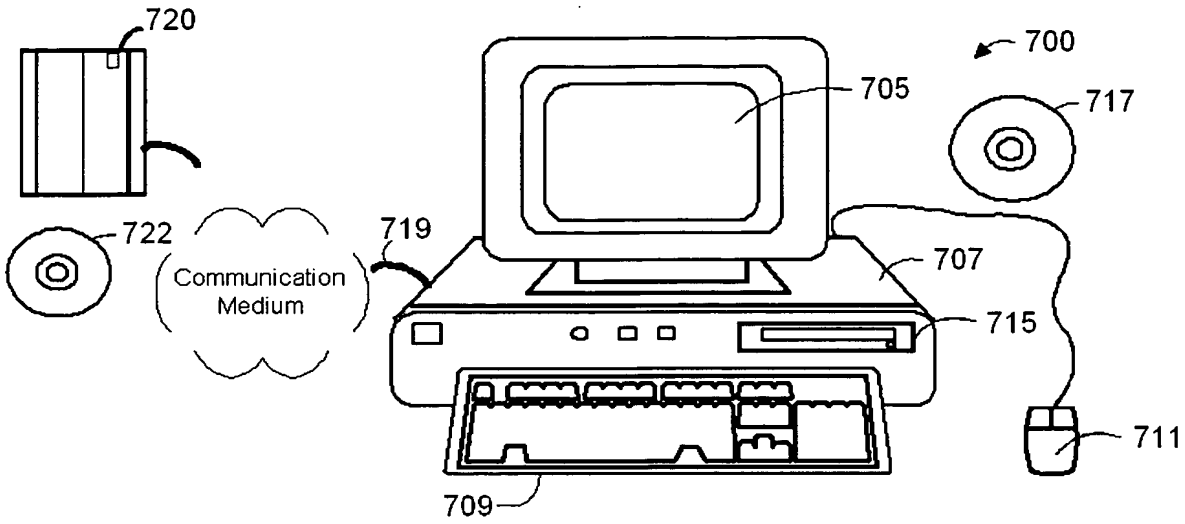


FIG. 8

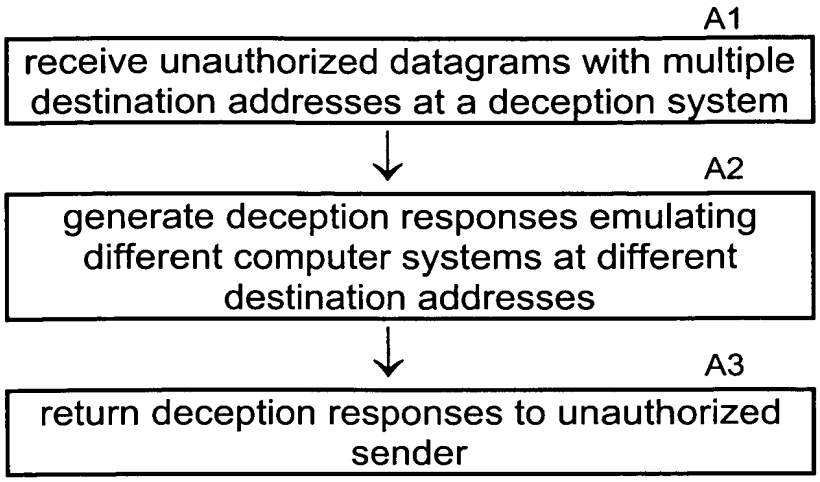

A1
receive unauthorized datagrams with multiple destination addresses at a deception system
A2
generate deception responses emulating different computer systems at different destination addresses
A3
return deception responses to unauthorized sender

FIG. 9

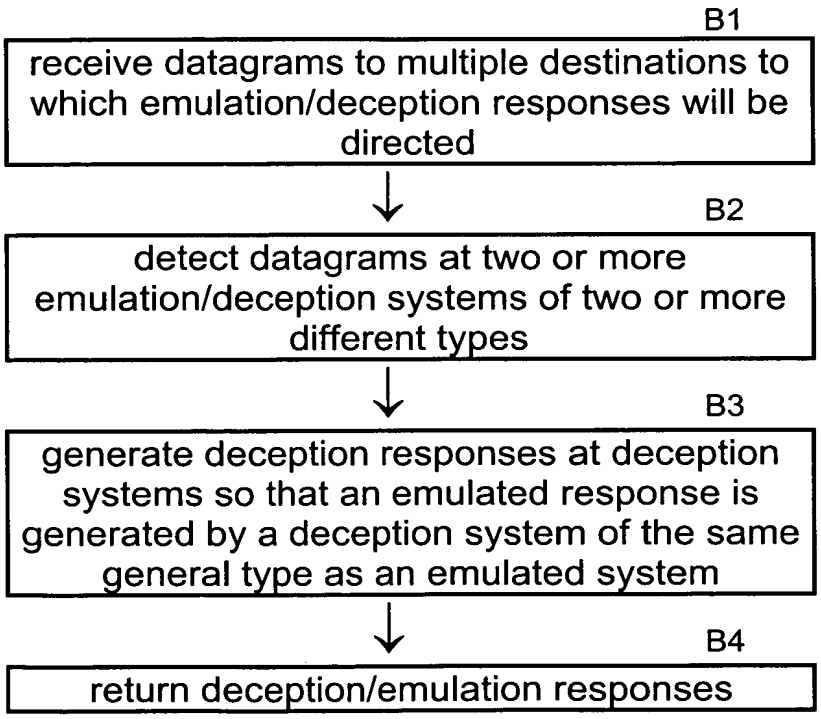

B1
receive datagrams to multiple destinations to which emulation/deception responses will be directed
B2
detect datagrams at two or more emulation/deception systems of two or more different types
B3
generate deception responses at deception systems so that an emulated response is generated by a deception system of the same general type as an emulated system
B4
return deception/emulation responses

FIG. 10

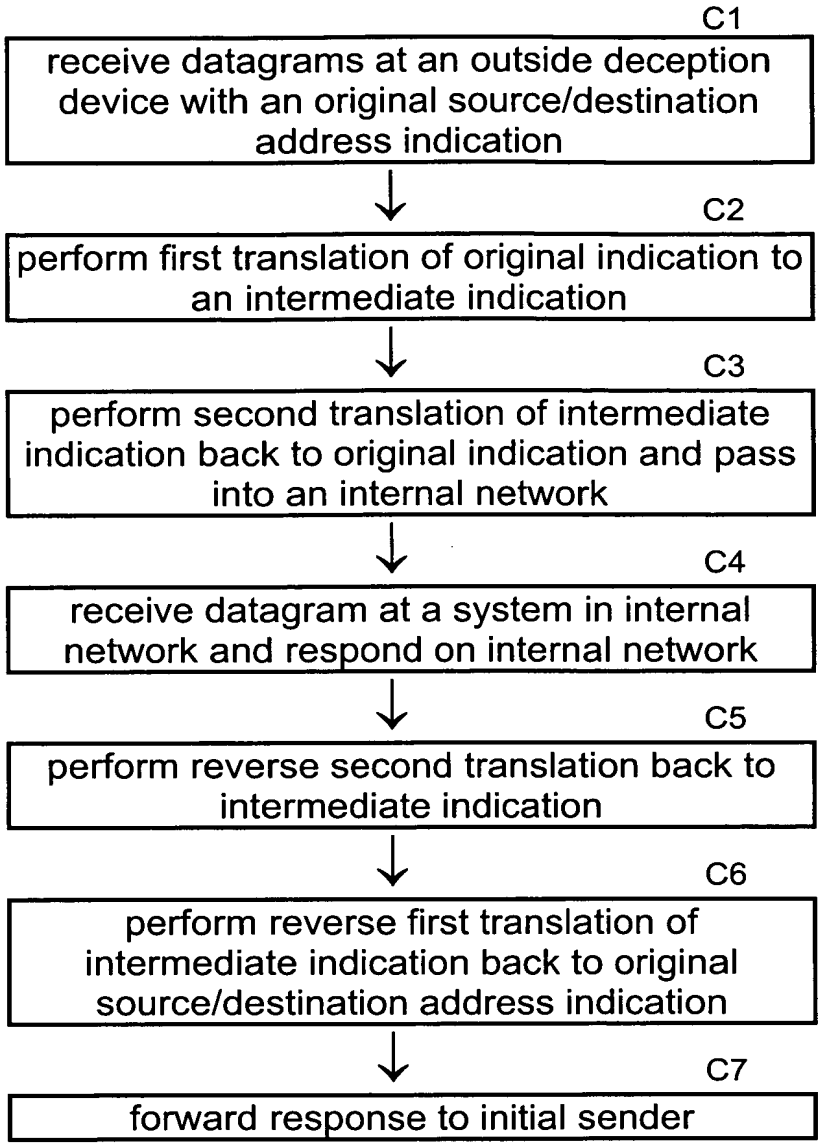

C1
receive datagrams at an outside deception device with an original source/destination address indication
C2
perform first translation of original indication to an intermediate indication
C3
perform second translation of intermediate indication back to original indication and pass into an internal network
C4
receive datagram at a system in internal network and respond on internal network
C5
perform reverse second translation back to intermediate indication
C6
perform reverse first translation of intermediate indication back to original source/destination address indication
C7
forward response to initial sender

FIG. 11

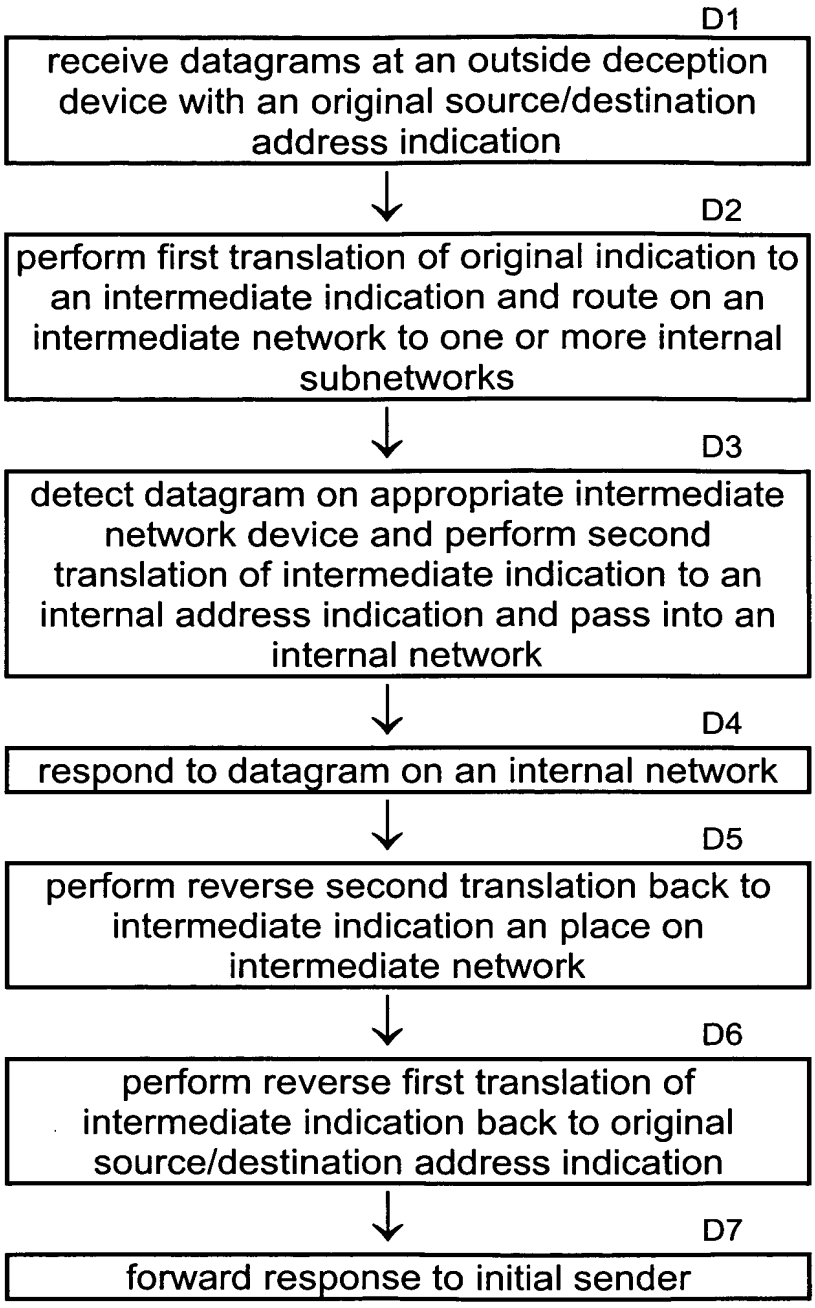

D1
receive datagrams at an outside deception device with an original source/destination address indication
D2
perform first translation of original indication to an intermediate indication and route on an intermediate network to one or more internal subnetworks
D3
detect datagram on appropriate intermediate network device and perform second translation of intermediate indication to an internal address indication and pass into an internal network
D4
respond to datagram on an internal network
D5
perform reverse second translation back to intermediate indication an place on intermediate network
D6
perform reverse first translation of intermediate indication back to original source/destination address indication
D7
forward response to initial sender

FIG. 12

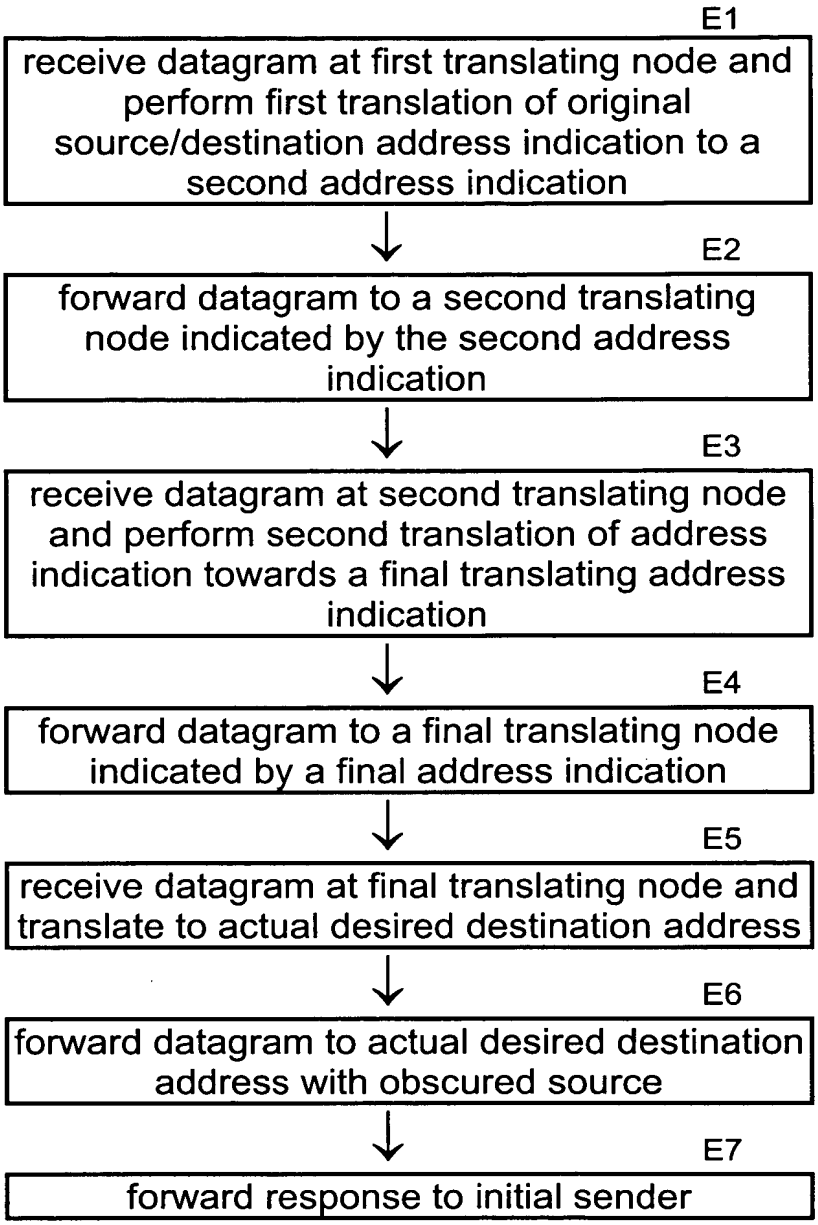

E1
receive datagram at first translating node and perform first translation of original source/destination address indication to a second address indication
E2
forward datagram to a second translating node indicated by the second address indication
E3
receive datagram at second translating node and perform second translation of address indication towards a final translating address indication
E4
forward datagram to a final translating node indicated by a final address indication
E5
receive datagram at final translating node and translate to actual desired destination address
E6
forward datagram to actual desired destination address with obscured source
E7
forward response to initial sender

FIG. 13

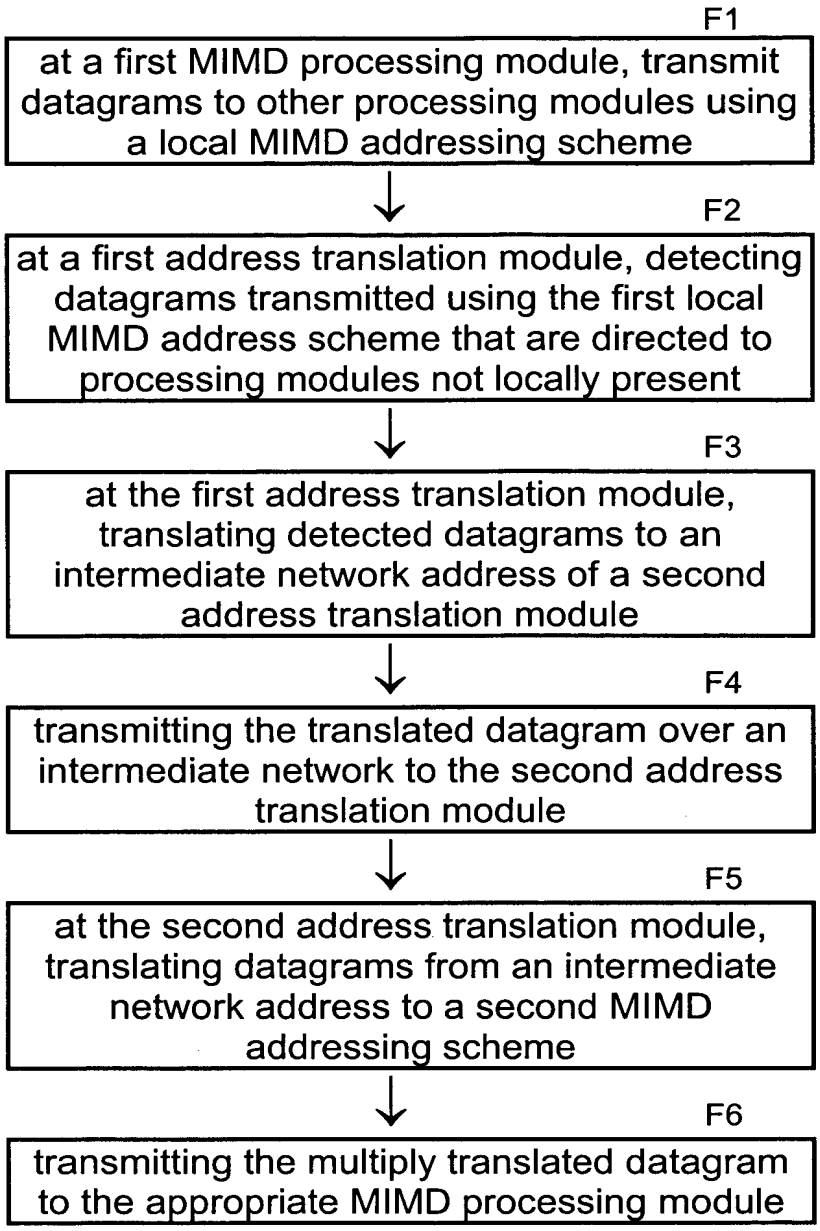

F1
at a first MIMD processing module, transmit datagrams to other processing modules using a local MIMD addressing scheme
F2
at a first address translation module, detecting datagrams transmitted using the first local MIMD address scheme that are directed to processing modules not locally present
F3
at the first address translation module, translating detected datagrams to an intermediate network address of a second address translation module
F4
transmitting the translated datagram over an intermediate network to the second address translation module
F5
at the second address translation module, translating datagrams from an intermediate network address to a second MIMD addressing scheme
F6
transmitting the multiply translated datagram to the appropriate MIMD processing module

FIG. 14